

02/09/24

Date: ☐☐☐/☐☐☐☐☐☐☐

Placement class

ASB-29

Q No.

Exercise No. ☐☐

Subjective Problems : Level / Section - ☐

DevOps Curriculum using with Tools.

Unit-1 DevOps workflow

1.1 Introduction to DevOps :

1.1.1. Definition and goals of DevOps

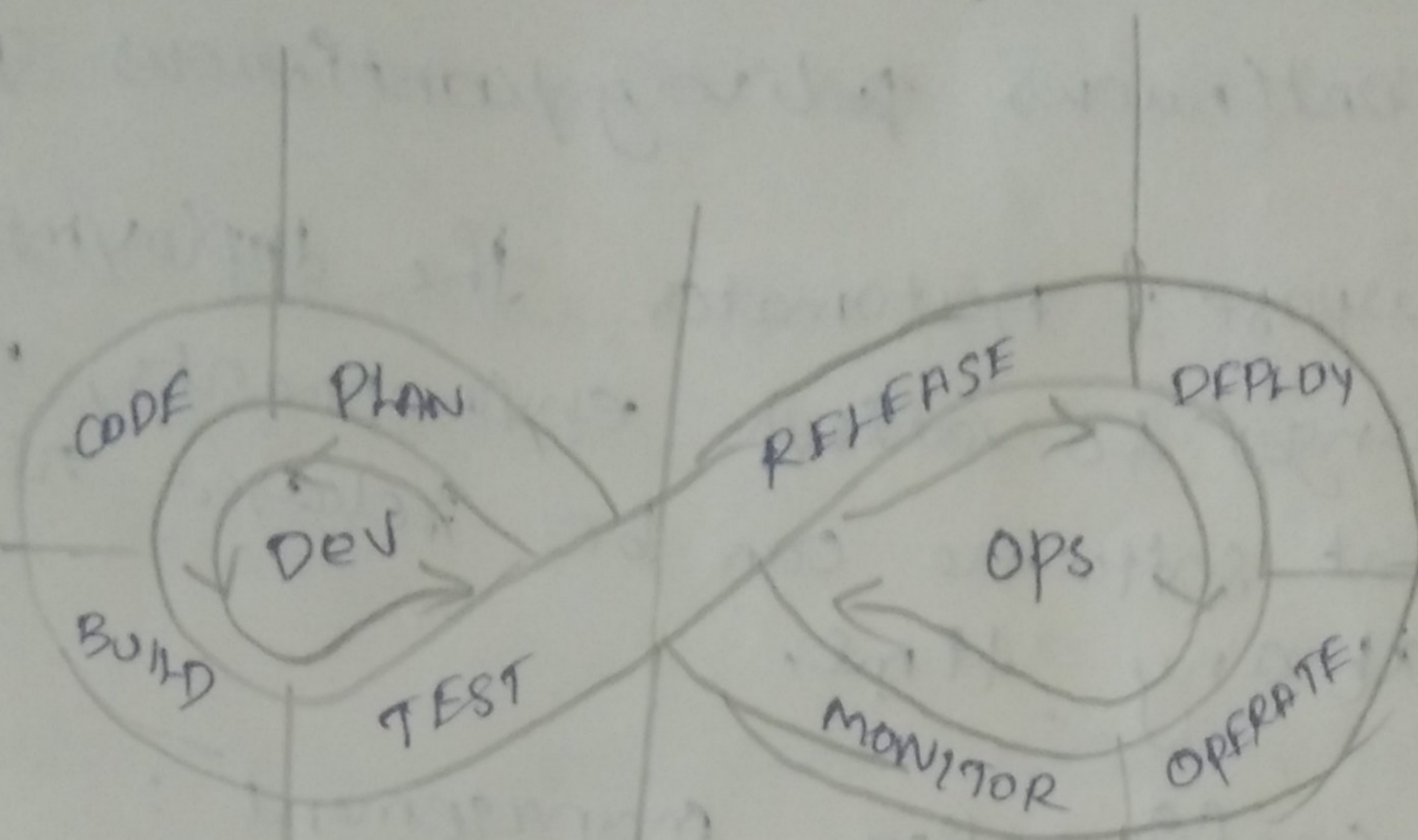
1.1.2 DevOps Architecture

1.1.3. DevOps Architecture workflow

1.1.1 Definition and goals DevOps

The main goal of DevOps are to improve the speed, efficiency, and quality of software development and delivery. Here are the primary objectives :

- * Increase Deployment Frequency
- * Improve Deployment Quality
- * Reduce Lead Time for changes
- * Enhance Collaboration and Communication
- * Improve Recovery Time
- * Automate and streamline processes



Team 1 - Plan, Code,

Team 2 - Build, Test

Team 3 - Release, Deploy

Team 4 - Operator, Monitor

1.1.2 - DevOps Architecture

Key components of DevOps Architecture:

* Version Control System (VCS):

Purpose: manages code versions, tracks changes, and facilitates collaboration among developers.

* Continuous Integration (CI):

Purpose: Automates the process of integrating code changes from multiple contributors into a single software project.

Date: / /

Q No.

Exercise No. Subjective Problems : Level / Section -

* Continuous Delivery / Continuous Deployment (CD):

Purpose: Automate the deployment of code changes to various environments, ensuring that software can be released reliably at any time.

* Configuration management:

Purpose: - manages and maintains consistency of software environments (development, testing, production).

* Infrastructure as code (IaC):

Purpose: manages and provisions computing infrastructure through machine-readable definition files, rather than physical hardware or interactive configuration tools.

* Containerization and Orchestration:

Purpose: packages applications and their dependencies into containers to ensure consistency across environments and simplifies deployment.

* Continuous monitoring and logging:

Purpose: monitors applications and infrastructure to detect performance issues, errors, and security threats.

* Collaboration and communication tools:-

Purpose: facilitates communication and collaboration among team members enabling faster decision-making and issue resolution.

1.1.3 DevOps workflow

Code: Developers write and commit code to a Version Control system (e.g., Git).

Build: The CI server automatically builds the code into executable files, creating artifacts that can be deployed.

Test: Automated tests are run to ensure the quality of the code. This includes unit tests, integration tests, and sometimes security checks.

Release: If all tests pass, the code is packaged and prepared for deployment.

Deploy:- The code is automatically deployed to the target environment (e.g., Staging, Production). Continuous Deployment involves deploying to production automatically, whereas Continuous Delivery might require manual approval.

Operate:- The deployed applications are monitored for performance, reliability, and security. Continuous monitoring tools collect metrics and logs, providing insights into the application's behaviour.

Monitor:-

Feedback is collected from monitoring and users, providing data for continuous improvements. Any issues detected are fed back into the development process for resolution.

1.2 DevOps Vs Traditional IT Operations

1.2.1 Differences between DevOps and traditional software development and IT operations.

1.2.2 Benefits of adopting Devops practices

- 1.2.3 Building a culture of collaboration and communication between development and Operations teams.
- 1.2.4 The role of automation and monitoring in enhancing team efficiency.
- 1.2.1 Differences between Devops and traditional software development and IT operations.

* Collaboration and communication:

* Traditional Approach: Development and IT Operations teams work in silos. Developers focus on writing code, and operations teams are responsible for deploying and maintaining the application. This often leads to miscommunication, delays, and a lack of shared understanding.

* Devops Approach: Devops encourages continuous collaboration and communication between development lifecycle, fostering a culture of shared responsibility.

Date: / /

Cl No:

Exercise No. /

Subjective Problems: Level / Section

Process to workflow:

Traditional Approach: uses a sequential development process (e.g. waterfall model) where each phase must be completed before the next begins. This can create bottlenecks and slow down the process.

Devops Approach: follows an agile and iterative approach where development, testing, and deployment are done continuously and concurrently. This helps identify and fix issues earlier in the development

Waterfall model:

It can make your projects flow smoothly, avoid bottlenecks, help you hit deadlines, ensure deliverables are not before the next phase begins, and allow the team overall to shine with perfection. This in-guide analyses the advantages of the waterfall methodology.

1. Requirement Gathering & Analysis
2. System ~~to~~ design
3. Implementation
4. Testing
5. Development
6. maintenance.

Agile:-

Agile development is important because it helps to ensure that development teams complete projects on time and within budget. It also helps to improve communication between the development team and the product owner. Additionally, Agile development methodology can help reduce the risks associated with complex projects.